What is claimed is:

CLAIMS

1. A method for issuing instructions in a processor having a pipeline, comprising:

(a) providing a loop buffer for holding program loop instructions and a register file for holding speculative and architectural loop control parameters;

(b) in response to decoding of a first loop setup instruction, marking a first entry in the register file as a current entry and writing in the first entry loop control parameters represented in the first loop setup instruction;

(c) marking the current entry in the register file as an architectural entry in response to the first loop setup instruction being committed in the pipeline;

(d) sending a loop bottom indicator down the pipeline with a loop bottom instruction.

2. A method as defined in claim 1, further comprising decrementing a loop count in the architectural entry in the register file in response to the loop bottom instruction being committed in the pipeline.

3. A method as defined in claim 1, further comprising issuing instructions of the program loop according to the loop control parameters in the current entry in the register file.

4. A method as defined in claim 1, wherein the register file has at least three entries.

5. A method as defined in claim 4, further comprising generating a current pointer to the current entry in the register file and generating an architectural pointer to the architectural entry in the register file.

6. A method as defined in claim 5, further comprising incrementing the current pointer to a second entry in the register file in response to decoding of a second loop setup instruction and writing in the second entry loop control parameters represented in the second loop setup instruction.

7. A method as defined in claim 6, further comprising incrementing the architectural pointer to the second entry in the register file in response to the second loop setup instruction being committed.

8. A method as defined in claim 6, further comprising moving the current pointer to a location of the architectural pointer in response to an interrupt or a pipeline abort.

9. A method as defined in claim 1, wherein step (b) comprises writing a loop top address to a loop top register, writing a loop bottom address to a loop bottom register and writing a loop count to a loop count register.

10. A method as defined in claim 9, further comprising comparing a current instruction address with the loop top address to determine a loop top match and comparing the current instruction address with the loop bottom address to determine a loop bottom match.

11. A method as defined in claim 10, further comprising writing a temporary loop count in a temporary loop count register and decrementing the temporary loop count on each loop bottom match.

12. A method as defined in claim 11, further comprising exiting the program loop when the temporary loop count has decremented to zero.

13. A method as defined in claim 1, further comprising stalling a loop setup instruction when the register file does not have an available entry.

14. A method as defined in claim 1, wherein instructions are issued without sending the loop control parameters down the pipeline.

15. A method as defined in claim 1, further comprising writing instructions of the program loop to the loop buffer on a first iteration of the program loop.

16. A method for controlling a program loop in a processor having a pipeline, comprising:

(a) providing a loop buffer for holding program loop instructions and a register file having at least three entries for holding speculative and architectural loop control parameters;

(b) marking a first entry in the register file as a current entry in response to decoding of a first loop setup instruction and writing in the first entry loop control parameters represented in the first loop setup instruction; and

(c) marking the first entry in the register file as an architectural entry in response to the first loop setup instruction being committed in the pipeline.

17. Apparatus for issuing instructions in a processor having a pipeline, comprising:

a loop buffer for holding program loop instructions;

a register file having at least three entries for holding speculative and architectural loop control parameters; and

a controller including means for marking a first entry in the register file as a current entry in response decoding of a first loop setup instruction and for writing in the first entry loop control parameters represented in the first loop setup instruction, and means for marking the current entry in the register file as an architectural entry in response to the first loop setup instruction being committed.

18. Apparatus as defined in claim 17, wherein the controller further comprises means for issuing instructions of the program loop according to the loop control parameters in the current entry in the register file, sending a loop bottom indicator down the pipeline with a loop bottom instruction, and decrementing a loop count in the architectural entry in the register file in response to the loop bottom instruction being committed in the pipeline.

19. Apparatus as defined in claim 18, wherein the controller further comprises means for marking a second entry in the register file as the current entry in response to decoding of a second loop setup instruction and for writing in the second entry loop control parameters represented in the second loop setup instruction, and means for marking the second entry in

the register file as the architectural entry in response to the second loop setup instruction being committed.

20. Apparatus as defined in claim 17, wherein each entry in the register file comprises a loop top register for holding a loop top address, a loop bottom register for holding a loop bottom address and a loop count register for holding a loop count.

21. Apparatus as defined in claim 20, further comprising a loop top comparator for comparing a current instruction address with the loop top address to determine a loop top match and a loop bottom comparator for comparing the current instruction address with the loop bottom address to determine a loop bottom match.

22. Apparatus as defined in claim 21, further comprising a temporary loop count register for holding a temporary loop count, wherein the controller further comprises means for decrementing the temporary loop count on each loop bottom match.

23. Apparatus as defined in claim 22, wherein the controller further comprises means for exiting the program loop when the temporary loop count has decremented to zero.

24. Apparatus as defined in claim 23, wherein the controller further comprises means for stalling when a loop setup instruction is decoded and the register set does not have an available entry.

25. Apparatus as defined in claim 17, wherein the controller is configured for operation without sending the loop control parameters down the pipeline.

26. Apparatus as defined in claim 17, wherein the controller includes means for writing instructions of the program loop to the loop buffer on a first loop iteration.

27. Apparatus as defined in claim 18, wherein the controller includes means for generating a current pointer for marking the current entry in the register file and for generating an architectural pointer for marking the architectural entry in the register file.

28. Apparatus as defined in claim 27, wherein the controller includes means for incrementing the current pointer in response to decoding of a loop setup instruction.

29. Apparatus as defined in claim 28, wherein the controller includes means for incrementing the architectural pointer in response to a loop setup instruction being committed in the pipeline.

30. Apparatus as defined in claim 27, wherein the controller includes means for moving the current pointer to a location of the architectural pointer in response to an interrupt or a pipeline abort.